Project topic: Improve the student performance in exams

Data source: The dataset, Student Performance in Exams, is from https://www.kaggle.com/spscientist/students-performance-in-exams

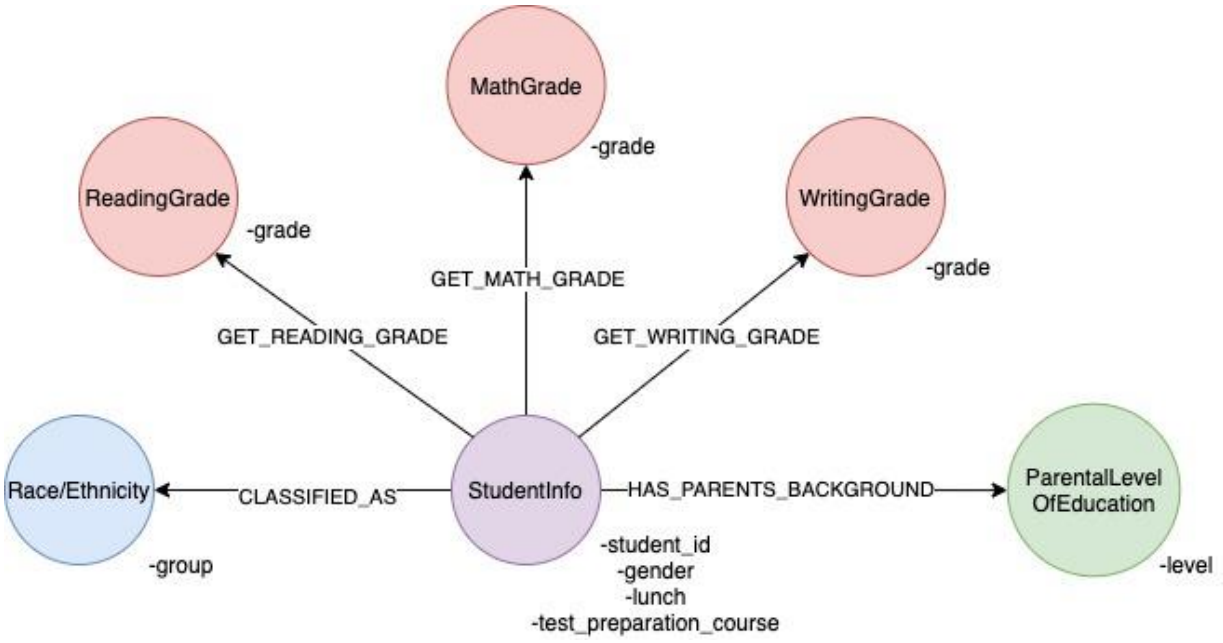
Goal

A high school chancellor would like to know the population about the students with their different social groups. Additionally, the chancellor also wants to know how to improve their students' exam performance in order to make the school better. To achieve this goal, I will use the student performance data which includes the students' past performance on three different subjects, and three factors, family, personal, and economics. Based on this information from the dataset, I am going to present graph analysis and identify the following questions,

- How do the three factors affect student's performance?
- How effective is the test preparation course?

This dataset contains 8 columns and 1000 rows. The first question can help me identify whether those factors are related to the performance. The second question can help find out if the existing preparation course is essential for students or the course should be improved so the student will not waste their time on it.
After knowing how the factors affect, the school might change the strategy of teaching or provide another specific course for student in order to have a better learning.

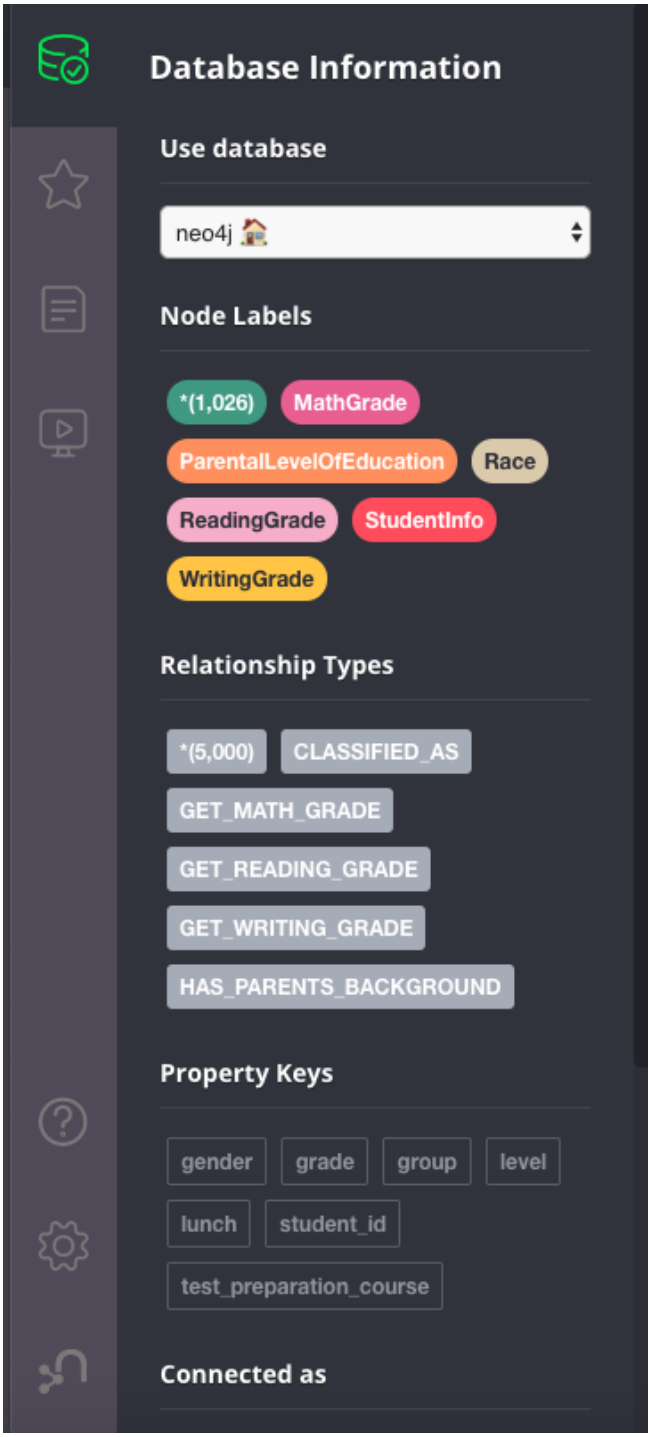
Graph data model



Data preprocessing

This dataset includes the student's test score on three subjects. Added the letter grade columns on each score in order to easier compare how the student's performance in each subject.

Neo4j database screenshot



Cypher queries

1. Overall looks for students who has at least one good grade, which is A or B, in all three subjects.

Code:

```
MATCH (s:StudentInfo)
WITH s
MATCH (s)-[:GET_MATH_GRADE]->(m:MathGrade)
MATCH (s)-[:GET_READING_GRADE]->(read:ReadingGrade)
MATCH (s)-[:GET_WRITING_GRADE]->(w:WritingGrade)
MATCH (s)-[:CLASSIFIED_AS]->(race:Race)
MATCH (s)-[:HAS_PARENTS_BACKGROUND]->(p:ParentalLevelOfEducation)
WHERE (m.grade = 'A' or m.grade = 'B') OR (read.grade = 'A' or read.grade = 'B') OR (w.grade = 'A' or w.grade = 'B')
RETURN m.grade AS good_math_grade, read.grade AS good_read_grade, w.grade AS good_writing_grade, s.lunch AS lunch_status, s.test_preparation_course AS test_preparation_course, s.gender AS gender, race.group AS race, p.level AS parental_level_of_edu
```

Result table:

	good_math_grade	good_read_grade	good_writing_grade	lunch_status	test_preparation_course	gender	race	parental_level_of_edu
70	"B"	"C"	"D"	"standard"	"none"	"male"	"group E"	"high school"
71	"A"	"C"	"D"	"standard"	"none"	"male"	"group C"	"high school"
72	"C"	"A"	"B"	"standard"	"completed"	"female"	"group C"	"bachelor's degree"
73	"A"	"A"	"B"	"standard"	"completed"	"male"	"group A"	"some college"
74	"C"	"A"	"B"	"standard"	"none"	"female"	"group C"	"some high school"
75	"D"	"A"	"B"	"standard"	"completed"	"female"	"group C"	"some college"
76	"A"	"A"	"B"	"free/reduced"	"none"	"male"	"group E"	"associate's degree"

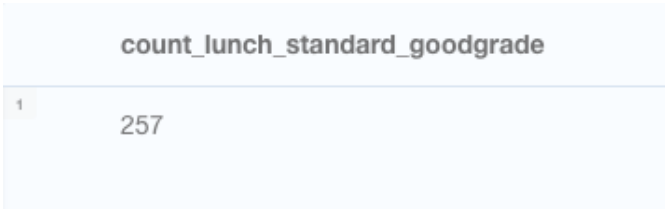
ted streaming 327 records after 3 ms and completed after 154 ms.

2. Count how many students who get at least one good grade with ‘standard’ lunch status

Code:

```
MATCH (s:StudentInfo)
WITH s
MATCH (s)-[:GET_MATH_GRADE]->(m:MathGrade)
MATCH (s)-[:GET_READING_GRADE]->(read:ReadingGrade)
MATCH (s)-[:GET_WRITING_GRADE]->(w:WritingGrade)
WHERE (m.grade = 'A' or m.grade = 'B' OR read.grade = 'A' or read.grade = 'B' OR w.grade = 'A'
or w.grade = 'B') AND (s.lunch = 'standard')
RETURN count(s.lunch) AS count_lunch_standard_goodgrade
```

Result table:



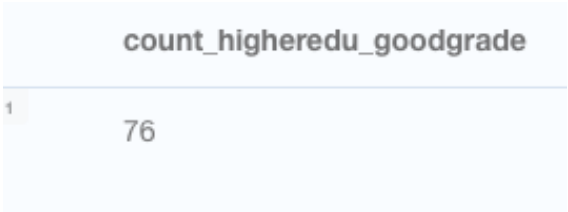
	count_lunch_standard_goodgrade
1	257

3. Count how many students whose parent's have higher education and get at least one good grade in the exams

Code:

```
MATCH (s:StudentInfo)
WITH s
MATCH (s)-[:GET_MATH_GRADE]->(m:MathGrade)
MATCH (s)-[:GET_READING_GRADE]->(read:ReadingGrade)
MATCH (s)-[:GET_WRITING_GRADE]->(w:WritingGrade)
MATCH (s)-[:HAS_PARENTS_BACKGROUND]->(p:ParentalLevelOfEducation)
WHERE (m.grade = 'A' or m.grade = 'B' OR read.grade = 'A' or read.grade = 'B' OR w.grade = 'A'
or w.grade = 'B') AND (p.level = "master's degree" or p.level = "bachelor's degree")
RETURN count(p.level) AS count_higheredu_goodgrade
```

Result table:



	count_higheredu_goodgrade
1	76

Application of Algorithms

Projection: Created a cypher projection of the students who shared same reading and writing grade.

Code:

```
CALL gds.graph.create.cypher('readwrite'),
'MATCH (s1:StudentInfo) RETURN id(s1) AS id',
'MATCH (s1:StudentInfo)-[:GET_READING_GRADE | :GET_WRITING_GRADE]->(read_write)<-[:GET_READING_GRADE | :GET_WRITING_GRADE]-(s2:StudentInfo) RETURN id(s1) AS source, id(s2) AS target')
```

1. Louvain algorithm

First, I used the 'readwrite' projection I created, applied Louvain algorithm to see the communities of students who shared the same reading and writing grade. From the result table, we can see there are 4 communities.

Code:

```
CALL gds.louvain.stream('readwrite')
YIELD nodeId, communityId AS community
RETURN community, collect(gds.util.asNode(nodeId).student_id) AS student
ORDER BY size(student) DESC
```

Result table:

	community	student
1	34	[2, 3, 6, 7, 13, 17, 25, 35, 36, 37, 39, 50, 55, 57, 64, 78, 86, 87, 90, 95, 96, 103, 105, 107, 111, 115, 117, 118, 121, 122, 123, 125, 126, 129, 131, 134, 147, 149, 150, 156, 159, 162, 165, 166, 169, 172, 173, 176, 180, 190, 192, 195, 201, 202, 208, 209, 215, 216, 217, 222, 224, 230, 233, 234, 235, 240, 242, 252, 254, 260, 264, 267, 269, 275, 277, 279, 283, 286, 287, 288, 295, 300, 301, 303, 305, 307, 311, 317, 321, 322, 323, 326, 334, 335, 346, 348, 353, 360, 362, 370, 374, 377, 378, 379, 381, 382, 383, 398, 404, 408, 410, 412, 421, 424, 427, 440, 441, 442, 444, 445, 448, 451, 452, 457, 459, 464, 466, 469, 471, 473, 475, 476, 490, 491, 493, 494, 501, 502, 504, 506, 510, 515, 516, 517, 522, 534, 536, 540, 542, 544, 546, 547, 552, 563, 564, 567, 572, 575, 578, 581, 582, 584, 595, 600, 603, 605, 607, 613, 615, 618, 619, 624, 626, 633, 634, 635, 637, 638, 642, 643, 644, 646, 650, 652, 653, 660, 668, 674, 678, 686, 690, 693, 696, 697, 701, 703, 709, 711, 712, 713, 714, 716, 718, 720, 723, 729, 731, 733, 737, 743, 750, 754, 756, 763, 779, 780, 782, 783, 785, 794, 798, 802, 803, 804, 806, 814, 815, 816, 820, 821, 822, 824, 829, 832, 834, 838, 846, 847, 853, 856, 858, 862, 864, 865, 866, 873, 874, 877, 879, 886, 887, 891, 892, 900, 901, 904, 905, 908, 912, 917, 919, 920, 935, 939, 942, 947, 952, 957, 958, 960, 963, 970, 971, 976, 980, 983, 984, 985, 990, 991, 992, 996, 1000]
2	7	[4, 8, 10, 11, 12, 15, 18, 19, 20, 23, 27, 34, 40, 43, 45, 46, 51, 53, 56, 58, 59, 60, 62, 65, 67, 69, 72, 73, 74, 75, 76, 77, 81, 82, 83, 85, 92, 94, 104, 112, 113, 114, 124, 130, 132, 136, 137, 138, 143, 144, 146, 154, 158, 163, 164, 167, 175, 177, 183, 185, 189, 196, 198, 199, 205, 212, 214, 218, 220, 226, 228, 232, 238, 239, 243, 244, 250, 251, 255, 263, 266, 272, 273, 280, 281, 282, 285, 293, 297, 299, 302, 308, 310, 312, 324, 325, 328, 330, 331, 332, 333, 338, 339, 340, 352, 354, 363, 364, 365, 366, 367, 376, 384, 385, 394, 396, 400, 401, 402, 405, 409, 419, 420, 422, 423, 425, 429, 433, 434, 436, 439, 449, 454, 456, 458, 461, 462, 465, 467, 482, 484, 485, 487, 489, 497, 505, 508, 512, 513, 521, 523, 524, 525, 526, 528, 529, 532, 537, 553, 555, 556, 565, 566, 569, 576, 577, 579, 591, 597, 598, 601, 602, 604, 608, 611, 612, 617, 623, 625, 628, 629, 630, 632, 641, 649, 651, 659, 666, 672, 683, 684, 689, 691, 707, 708, 722, 724, 725, 728, 730, 732, 734, 735, 736, 740, 742, 745, 749, 755, 757, 759, 762, 770, 771, 776, 778, 781, 784, 786, 788, 795, 799, 800, 808, 810, 811, 812, 817, 823, 825, 830, 831, 833, 835, 839, 841, 842, 843, 848, 859, 863, 867, 868, 870, 871, 884, 885, 890, 896, 897, 899, 903, 907, 911, 922, 927, 929, 930, 932, 937, 938, 948, 949, 954, 956, 961, 962, 965, 973, 979, 981, 986, 987, 989, 997]
3	20	[9, 21, 28, 29, 32, 33, 38, 41, 42, 44, 47, 52, 63, 66, 70, 71, 80, 84, 89, 97, 99, 100, 101, 106, 108, 110, 119, 120, 127, 139, 140, 142, 148, 151, 152, 153, 155, 160, 174, 179, 181, 182, 184, 186, 187, 188, 191, 193, 194, 197, 203, 204, 207, 210, 213, 221, 223, 225, 229, 231, 237, 241, 245, 248, 249, 257, 265, 271, 278, 291, 296, 298, 304, 306, 309, 313, 314, 315, 316, 319, 320, 327, 336, 337, 341, 342, 344, 345, 347, 349, 351, 357, 358, 359, 361, 368, 369, 380, 388, 389, 391, 392, 393, 397, 399, 403, 406, 407, 413, 414, 416, 417, 418, 426, 428, 430, 431, 435, 437, 438, 443, 447, 455, 460, 468, 472, 474, 477, 478, 479, 480, 481, 483, 492, 495, 496, 499, 503, 507, 514, 527, 531, 533, 535, 538, 539, 541, 545, 548, 549, 550, 551, 554, 558, 560, 570, 573, 574, 580, 588, 589, 590, 592, 593, 606, 609, 610, 616, 620, 621, 622, 627, 631, 636, 645, 647, 648, 656, 657, 658, 662, 663, 664, 665, 669, 671, 676, 680, 682, 685, 687, 695, 700, 702, 704, 705, 706, 710, 738, 748, 751, 752, 758, 761, 764, 765, 768, 773, 775, 789, 790, 791, 792, 793, 796, 801, 809, 813, 819, 826, 827, 828, 836, 837, 845, 849, 850, 851, 852, 854, 855, 857, 861, 875, 876, 878, 881, 889, 893, 895, 910, 913, 914, 915, 916, 918, 921, 923, 924, 925, 926, 928, 936, 940, 941, 943, 944, 945, 946, 955, 959, 967, 968, 969, 972, 974, 975, 977, 978, 995, 998]
4	4	[1, 5, 14, 16, 22, 24, 26, 30, 31, 48, 49, 54, 61, 68, 79, 88, 91, 93, 98, 102, 109, 116, 128, 133, 135, 141, 145, 157, 161, 168, 170, 171, 178, 200, 206, 211, 219, 227, 236, 246, 247, 253, 256, 258, 259, 261, 262, 268, 270, 274, 276, 284, 289, 290, 292, 294, 318, 329, 343, 350, 355, 356, 371, 372, 373, 375, 386, 387, 390, 395, 411, 415, 432, 446, 450, 453, 463, 470, 486, 488, 498, 500, 509, 511, 518, 519, 520, 530, 543, 557, 559, 561, 562, 568, 571, 583, 585, 586, 587, 594, 596, 599, 614, 639, 640, 654, 655,

2. Label Propagation

Second, I applied the different community algorithm, label propagation, to see how the communities be clustered. According to the result, the label propagation clustered three communities for the students who shared the same writing and reading grade.

Code:

```
CALL gds.labelPropagation.stream('readwrite’)
YIELD nodeId, communityId AS Community
RETURN Community, collect(gds.util.asNode(nodeId).student_id) AS student
ORDER BY size(student) DESC
```

Result table:

	Community	student
1	4	[1, 2, 3, 5, 6, 7, 13, 14, 16, 17, 22, 24, 25, 26, 28, 29, 30, 31, 33, 35, 36, 37, 39, 42, 48, 49, 50, 54, 55, 57, 61, 64, 68, 71, 78, 79, 86, 87, 88, 89, 90, 91, 93, 95, 96, 98, 102, 103, 105, 106, 107, 109, 110, 111, 115, 116, 117, 118, 121, 122, 123, 125, 126, 128, 129, 131, 133, 134, 135, 141, 145, 147, 149, 150, 153, 156, 157, 159, 161, 162, 165, 166, 168, 169, 170, 171, 172, 173, 174, 176, 178, 179, 180, 181, 184, 187, 190, 192, 195, 200, 201, 202, 203, 206, 208, 209, 211, 215, 216, 217, 219, 221, 222, 224, 225, 227, 229, 230, 233, 234, 235, 236, 240, 242, 245, 246, 247, 248, 252, 253, 254, 256, 257, 258, 259, 260, 261, 262, 264, 265, 267, 268, 269, 270, 274, 275, 276, 277, 278, 279, 283, 284, 286, 287, 288, 289, 290, 291, 292, 294, 295, 298, 300, 301, 303, 304, 305, 307, 309, 311, 315, 317, 318, 319, 321, 322, 323, 326, 329, 334, 335, 337, 342, 343, 344, 345, 346, 348, 349, 350, 353, 355, 356, 360, 362, 370, 371, 372, 373, 374, 375, 377, 378, 379, 381, 382, 383, 386, 387, 388, 390, 392, 395, 398, 404, 406, 408, 410, 411, 412, 415, 416, 417, 418, 421, 424, 427, 428, 432, 438, 440, 441, 442, 443, 444, 445, 446, 448, 450, 451, 452, 453, 457, 459, 463, 464, 466, 469, 470, 471, 472, 473, 475, 476, 478, 479, 480, 481, 486, 488, 490, 491, 492, 493, 494, 498, 500, 501, 502, 504, 506, 507, 509, 510, 511, 515, 516, 517, 518, 519, 520, 522, 530, 531, 534, 536, 539, 540, 542, 543, 544, 546, 547, 550, 551, 552, 557, 559, 561, 562, 563, 564, 567, 568, 571, 572, 575, 578, 581, 582, 583, 584, 585, 586, 587, 594, 595, 596, 599, 600, 603, 605, 606, 607, 613, 614, 615, 616, 618, 619, 620, 624, 626, 627, 631, 633, 634, 635, 636, 637, 638, 639, 640, 642, 643, 644, 646, 647, 650, 652, 653, 654, 655, 660, 661, 663, 667, 668, 669, 670, 671, 673, 674, 675, 677, 678, 679, 681, 686, 687, 688, 690, 692, 693, 694, 696, 697, 698, 699, 701, 702, 703, 706, 709, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 723, 726, 727, 729, 731, 733, 737, 738, 739, 741, 743, 744, 746, 747, 750, 752, 753, 754, 756, 758, 760, 761, 763, 766, 767, 769, 772, 773, 774, 777, 779, 780, 782, 783, 785, 787, 792, 793, 794, 796, 797, 798, 801, 802, 803, 804, 805, 806, 807, 809, 814, 815, 816, 818, 819, 820, 821, 822, 824, 827, 828, 829, 832, 834, 836, 838, 840, 844, 846, 847, 849, 850, 853, 856, 858, 860, 862, 864, 865, 866, 869, 872, 873, 874, 877, 879, 880, 882, 883, 886, 887, 889, 891, 892, 893, 894, 898, 900, 901, 902, 904, 905, 906, 908, 909]
2	7	[4, 8, 10, 11, 12, 15, 18, 19, 20, 23, 27, 34, 38, 40, 41, 43, 45, 46, 51, 53, 56, 58, 59, 60, 62, 63, 65, 67, 69, 70, 72, 73, 74, 75, 76, 77, 81, 82, 83, 85, 92, 94, 104, 112, 113, 114, 124, 130, 132, 136, 137, 138, 143, 144, 146, 154, 158, 163, 164, 167, 175, 177, 183, 185, 189, 196, 198, 199, 205, 212, 213, 214, 218, 220, 226, 228, 232, 238, 239, 243, 244, 250, 251, 255, 263, 266, 272, 273, 280, 281, 282, 285, 293, 296, 297, 299, 302, 308, 310, 312, 324, 325, 328, 330, 331, 332, 333, 338, 339, 340, 341, 347, 352, 354, 361, 363, 364, 365, 366, 367, 369, 376, 384, 385, 391, 394, 396, 399, 400, 401, 402, 403, 405, 409, 413, 419, 420, 422, 423, 425, 429, 430, 431, 433, 434, 436, 439, 447, 449, 454, 456, 458, 461, 462, 465, 467, 482, 484, 485, 487, 489, 497, 503, 505, 508, 512, 513, 521, 523, 524, 525, 526, 527, 528, 529, 532, 533, 537, 553, 555, 556, 565, 566, 569, 573, 576, 577, 579, 591, 592, 597, 598, 601, 602, 604, 608, 611, 612, 617, 621, 622, 623, 625, 628, 629, 630, 632, 641, 645, 649, 651, 659, 666, 672, 680, 682, 683, 684, 689, 691, 695, 700, 707, 708, 710, 722, 724, 725, 728, 730, 732, 734, 735, 736, 740, 742, 745, 749, 755, 757, 759, 762, 765, 768, 770, 771, 775, 776, 778, 781, 784, 786, 788, 791, 795, 799, 800, 808, 810, 811, 812, 817, 823, 825, 826, 830, 831, 833, 835, 837, 839, 841, 842, 843, 848, 855, 859, 861, 863, 867, 868, 870, 871, 878, 884, 885, 888, 890, 896, 897, 899, 903, 907, 911, 915, 918, 922, 926, 927, 929, 930, 932, 936, 937, 938, 946, 948, 949, 954, 956, 959, 961, 962, 965, 968, 973, 977, 978, 979, 981, 986, 987, 989, 997]
3	20	[9, 21, 32, 44, 47, 52, 66, 80, 84, 97, 99, 100, 101, 108, 119, 120, 127, 139, 140, 142, 148, 151, 152, 155, 160, 182, 186, 188, 191, 193, 194, 197, 204, 207, 210, 223, 231, 237, 241, 249, 271, 306, 313, 314, 316, 320, 327, 336, 351, 357, 358, 359, 368, 380, 389, 393, 397, 407, 414, 426, 435, 437, 455, 460, 468, 474, 477, 483, 495, 496, 499, 514, 535, 538, 541, 545, 548, 549, 554, 558, 560, 570, 574, 580, 588, 589, 590, 593, 609, 610, 648, 656, 657, 658, 662, 664, 665, 676, 685, 704, 705, 748, 751, 764, 789, 790, 813, 845, 851, 852, 854, 857, 875, 876, 881, 895, 913, 914, 923, 924, 940, 943, 944, 945, 955, 967, 974, 975, 995]

3. Node Similarity

Third, I applied node similarity to see how the students can be paired together by their reading and writing grade.
According to the results, we can know that students who get the similarity score 1.0 means that they have same reading and writing grade.

Code:

```
CALL gds.graph.create('read_write',['StudentInfo','ReadingGrade','WritingGrade'],'*')

CALL gds.nodeSimilarity.stream('read_write')
YIELD node1, node2, similarity
```

RETURN gds.util.asNode(node1).student_id AS Student1, gds.util.asNode(node2).student_id AS Student2, similarity
ORDER BY similarity DESC, Student1, Student2

Result table:

	Student1	Student2	similarity
8	1	31	1.0
9	1	48	1.0
10	1	49	1.0
11	2	39	1.0
12	2	103	1.0
13	2	121	1.0
14			

Cypher Actions

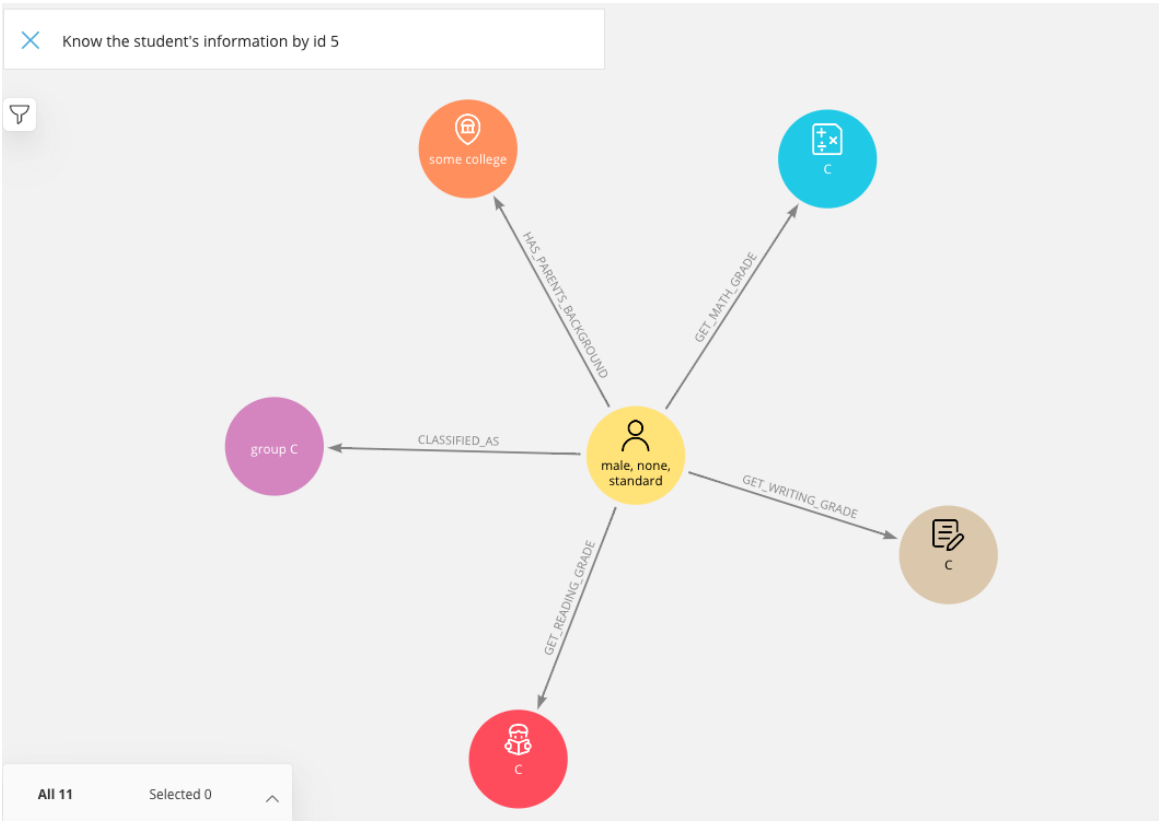
1. Know the student's information by id $student_id

This query can let user(may be the student's advisor) quickly search the student by id in order to know more details about the student with his/her grades or social status.

Code:

```
MATCH (s:StudentInfo{student_id:$student_id})
WITH s
MATCH (s)-[r1: GET_MATH_GRADE]-(m:MathGrade)
MATCH (s)-[r2:GET_READING_GRADE]-(read:ReadingGrade)
MATCH (s)-[r3:GET_WRITING_GRADE]-(w:WritingGrade)
MATCH (s)-[r4:HAS_PARENTS_BACKGROUND]-(p:ParentalLevelOfEducation)
MATCH (s)-[r5:CLASSIFIED_AS]-(race:Race)
RETURN s,m,r1,r2,r3,r4,r5,read,w,p,race
```

The example below, I search student ID = 5, the graph shows all the features that related to this student. This action is good for teachers who want to know about his student.

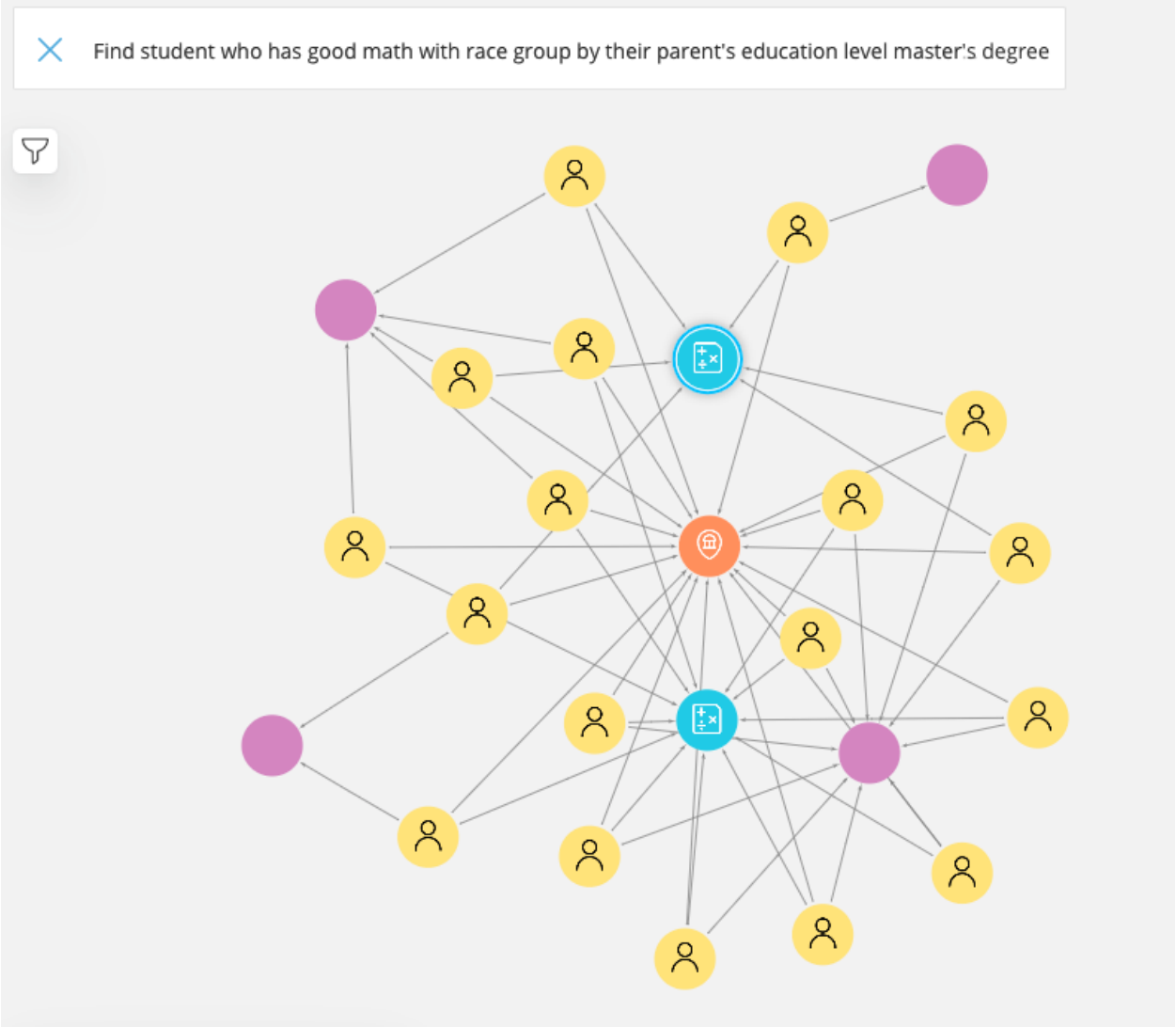


2. Find student who has good math with race group search by their parent's education level $level
This action can get multiple information base on the student who has good performance on math.

Code:

```
MATCH (s:StudentInfo)-
[r1:HAS_PARENTS_BACKGROUND]->(p:ParentalLevelOfEducation{level:$level})
WITH s,r1,p
MATCH (race:Race)<-[r2:CLASSIFIED_AS]-(s)
MATCH (s)-[r3:GET_MATH_GRADE]-(m:MathGrade)
WHERE m.grade = "A" OR m.grade = "B"
RETURN race,s,r1,p,r2,r3,m
```

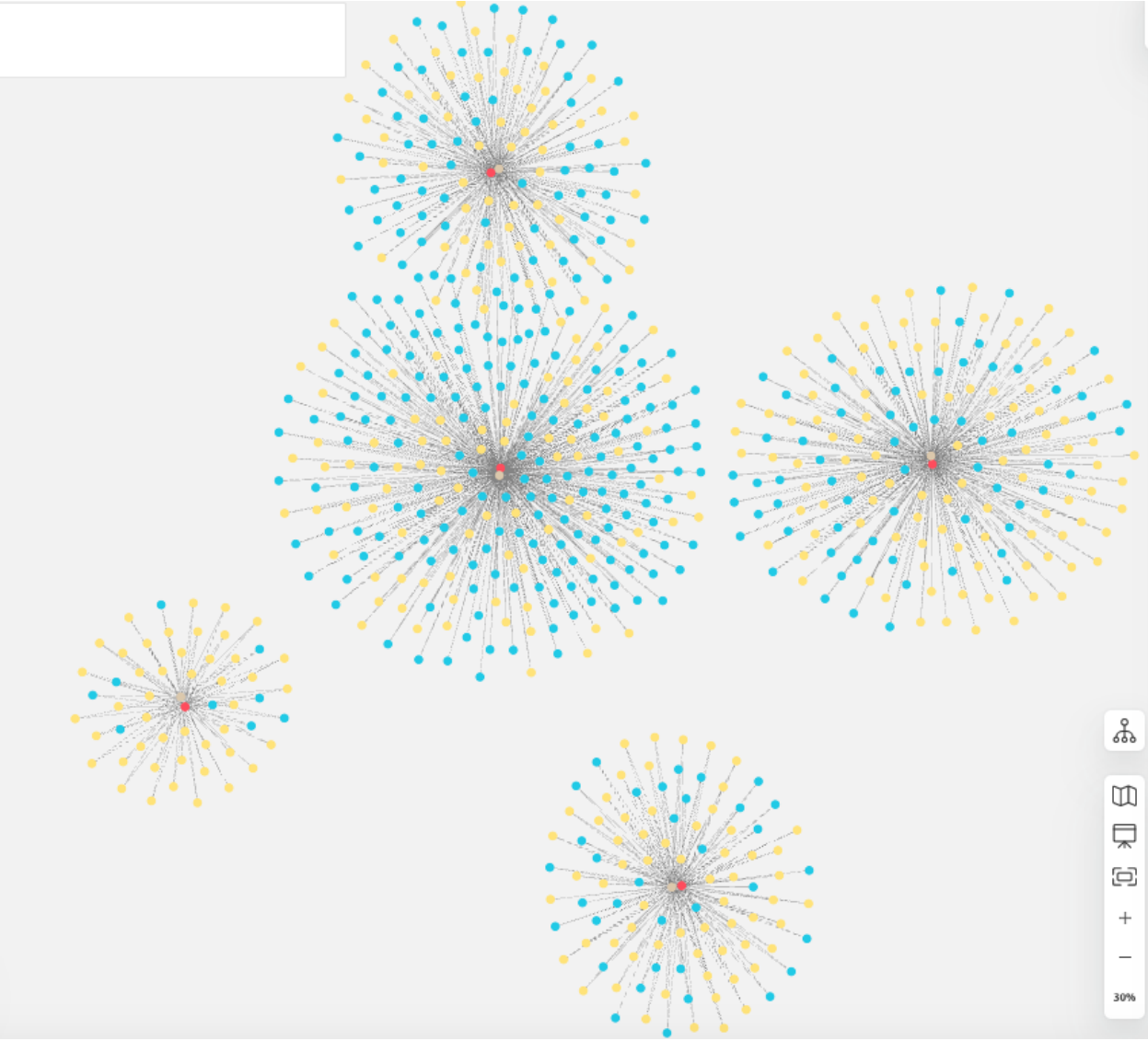
For example, when I search master's degree, we can easily find that there is no race group A in the image. The purple node on the upper right side, group B, has only one student match with good math grade and high parental education.



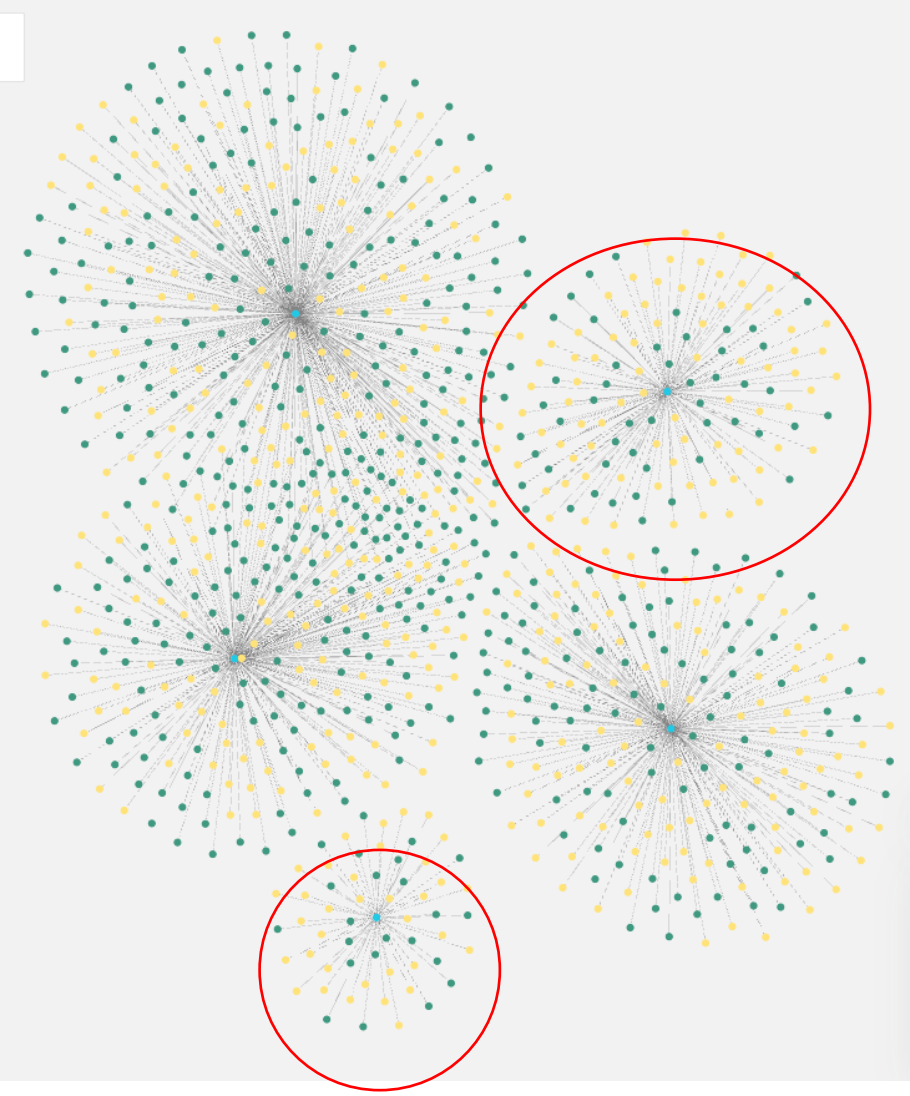
Visualization

1. Students who share the same reading grade and writing grade. The blue color means male and the yellow means female. In the middle circle, the students share grade F, which contains more male students. Additionally, it looks like if student who failed on one of reading or writing exam, most likely that student will also fail on the other one.
On the bottom left circle, that is grade A with more female students. Therefore, female students have better performance in the reading and writing fields. The school may provide more materials that can help boys improving their reading and writing grade.

Code:
```
match (s:StudentInfo)-[r2:GET_READING_GRADE]->(r:ReadingGrade)
match (s:StudentInfo)-[r3:GET_WRITING_GRADE]->(w:WritingGrade)
where r.grade = w.grade
Return *
```

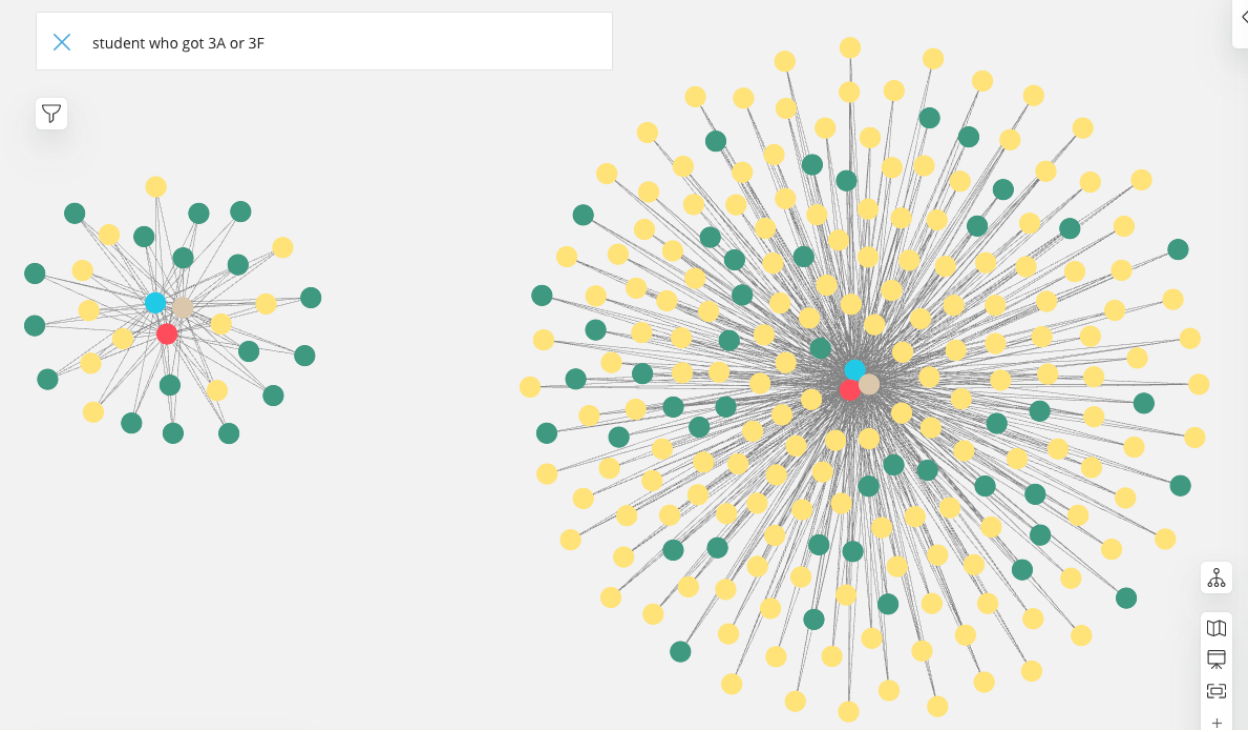


2. Based on the first insights, it seems like gender might be a factor which impacts on the student's exam performance. Therefore, I created another graph to visualize the math grade with applying the color on gender.
As the graph shown below, the two clusters which circled by red line, are A and B on math. The green nodes are represented as female students. Since the green nodes are less in these two grades and more in other three clusters, we can know that boys are more likely to have good performance in math.



3. Students who got all A or all F in three exams, comparing with their test preparation course status. In the image below, on the left side is all A and the right side is all F, with green nodes which means the student who have completed the course. According to this result, we can know that the student who got all A took more effort on the exams. Most of them finished the preparation course. On the other hand, most of the student who failed on all exams did not complete the course.

```
MATCH (m:MathGrade)<-[r1:GET_MATH_GRADE]-(s:StudentInfo)
MATCH (r:ReadingGrade)<-[r2:GET_READING_GRADE]-(s:StudentInfo)
MATCH (w:WritingGrade)<-[r3:GET_WRITING_GRADE]-(s:StudentInfo)
WHERE (m.grade = 'F' AND r.grade = 'F' AND w.grade = 'F') OR (m.grade = 'A' AND r.grade ='A'
AND w.grade = 'A')
RETURN *
```


student who got 3A or 3F

Summarize

In this analysis, I discovered information about this school's students their past performance on three subjects. I began by identifying the good performance in the exams, which is at least one A or B in any of the three exams. Then, I investigated the different factors, which include lunch status, parent's education level, race group, and gender, have an impact on student's grade or not.

By writing cypher queries, I got a table which has an overall look at those students. Based on the result table, there are 327 students who got at least one good grade in the exams.

To further investigate the students who are in the group of good performance, I first used the factor, lunch status, to see how the economic factor affects their grade. The result shows there are 257 students whose lunch status is standard. Therefore, I got 257/327 students who have good grades with a standard price of lunch. It seems like most of the students with good exam performance doesn't have an economic problem.

Second, I took parental level of education into account. According to the counted result, I got 76 students whose parents' have higher level of education, which included master's and bachelor's degree. Therefore, there are 76/327 students who has good grade with having a higher education family background. This proportion looks like even the students' parents have no higher education they still got a good performance in the exams.

To further understand how the students can group together base on their performance in exams, I applied the community algorithms which are Louvain and Label Propagation methods. First, I created the monopartite projection which is the student who shared the same reading and writing grade. In Louvain algorithm, it clustered into four groups. However, the result from label propagation is slightly different. It clustered into three groups, which is one group less than the Louvain did. Based on these results, the school can distribute the student into three or four groups. By providing different levels of courses to different levels of students, they may have a better course experience and get some improvement on the score.

Second, I applied the node similarity algorithm to see how the students can pair with each other. Different from the strategy above, the school can use this result to pair the student who should sit together. If their similarity is equal to 1, they have both the same reading and writing

grades. If they have both bad grades, then maybe the teacher can assign a student with good score to sit by their side. This way is to encourage the student to learn from each other. The student has good reading or writing can teach the student who is not good at it. When they are teaching other students, they are also reviewing at the same time.

To benefit the school's faculty and staff, I created two search phrases to let them easily get some information from the students. The first query is to let users quickly search students by their id in order to know more details about the student with his/her grades or social status. The second one I chose to discover the different insights on math grades. This one is to find a student who has good math with race group search by their parent's education level. It can get multiple information at once base on the student who has good performance on math.

To bring different findings of the data, I first created a graph about the students who share the same reading grade and writing grade. Then, I applied the gender factor to see if this factor affects their grade. As the result, female students have better performance in the reading and writing fields. After I got this result, I was interested in if male students have better performance in math. Therefore, I created the second graph which is math grade also with gender factor applied. The result shows male students did better in math than females. Thus, the school may provide more materials that can help boys improving their reading and writing grade and help girls with more math materials. Third, I created a graph which is students who got all A or all F in three exams and applied the test preparation course factor into the graph. According to the result, the school may reconsider should they keep this course or adjust the course content because it looks like there are 1/10 students who have completed the course still failed on all 3 exams.

Based on the above analysis, I would suggest the chancellor first consider the preparation course they currently provided. There is no obvious clue to see this course really help their students on the exams. It is better not providing such course to waste the students time also teachers' on the course which cannot help them learn better. Second, they can distribute the student base on their level so that teacher can teach them by using different levels of materials. This way will help the lower-level students receive a better learning. Lastly, due to the gender factor has an impact on their exams, teachers can provide more math materials for girls and

reading and writing materials for boys. I believe the school can improve their student's performance on exams and make school better by accepting the suggestions above.